2020 年度
早稲田大学大学院基幹理工学研究科情報理工・情報通信専攻　修士論文

パッシブイメージャ画像を用いた不審物検知手法の精度比較に関する研究

A Study on Accuracy Comparison of Suspicious Object Detection Methods Using Passive-imager Images

菅野　成希

（5119F026-2）

提出日： 2021.1.25
指導教員： 亀山渉教授
研究指導名: マルチメディア情報システム研究

目次

第1章　序論

1.1.　研究の背景

近年、テロの脅威が増していくにつれ、空港をはじめとする公共交通機関や 2021 年に控えるオリンピックなどの公共交通機関においてセキュリティチェックはますます重要になってきている。しかし、現在のセキュリティチェック方式は時間が著しくかかり、検査渋滞が発生しやすいという問題がある。

この問題を解決するためには、自動的に人の流れから数多くの人物を迅速に検査する必要がある。

近年、W 帯を使用するレーダにより人物の衣服の上からその人物所持する物体を画像化するセンシング・イメージング技術が確立されつつあり、また、画像認識の分野においてCNN（Convolutional Neural Network）の技術は急速な反転を続けている。

センシング・イメージング技術のうち、微弱な電波を計測覆われている物体の透過率を画像化する技術はパッシブイメージング技術と呼ばれる。

また、画像認識分野において画像中のそれぞれの画素が背景および対象物体のどの領域に属するかを判別するタスクをセマンティックセグメンテーションと呼び、セマンティックセグメンテーションで画像を領域分けすることで、対象物体の位置推定も合わせて可能となる。

そこで本研究では、W 帯を用いるパッシブイメージング技術により得られるパッシブイメージャ画像に対して CNN を用いたセマンティックセグメンテーションを行うことで画像中の物体を領域分けを行い不審物の位置特定を行うことで、人の流れから自動的に多くの人物を検査するシステムへの足掛かりとする。

1.2.　研究の目的

従来の CNN を用いたセマンティックセグメンテーションでは後述する U-Net や PSPNet を用いられる。そのため、U-Net[1]とその改良モデルおよび PSPNet[2]で実験データを学習させ、その精度を比較することでどのモデルが不審物検知のタスクに向いているかを検証することが、まず本研究の 1 個目の目的である。

また、従来のセマンティックセグメンテーションに用いられるデータセットに比べて、本研究の実験に用いる画像の枚数は全てで 1,008 枚と少なく、実験画像のノイズが大きいという問題がある。そこで実験データの少なさおよび実験画像のノイズの影響を最小化するために、データオーギュメンテーションの工夫および既存のモデルに改良を施すことで精度の向上を目指すのが本研究の 2 個目の目的となる。

1.3. 本論文の構成

本論文は以下の 6 章で構成されている。

第 1 章では、本研究の背景と目的、本論文の構成について記述している。

第 2 章では、セマンティックセグメンテーション、オブジェクト認識およびデータオーギュメンテーションにおける既存研究について記述している。

第 3 章では、既存のセマンティック・セグメンテーショ手法を用いて不審物検知の比較実験を行い、その結果と考察について記述している。

第 4 章では、データオーギュメンテーションとして Mixup を利用して比較実験を行い、その結果と考察について記述している。

第 5 章では、U-Net に Residual Blokc[5]および Dense Block[7]を用いた改良を加えて比較実験を行い、その結果と考察について記述している。

第 6 章では、本研究のまとめおよび今後の課題について記述している。

第2章　先行研究

2.1. セマンティックセグメンテーション

セマンティックセグメンテーションとは、画像中のそれぞれの画素が背景および対象物体のどの領域に属するかを判別し、画像を領域分けにするタスクのことである。

画像認識における CNN のオブジェクト認識技術の発達に伴い、セマンティックセグメンテーションにも CNN が応用されるようになった。

オブジェクト認識における CNN では Convolution と Max Pooling によってダウンサンプリンを行い特徴量を抽出する。そして、最終的に全結合層を設置し Softmax 関数を用いることによってどのオブジェクトであるかを判別する構造を持つ。

それに対し、セマンティックセグメンテーションでは、Convolution と Max Pooling によるダウンサンプリンによって特徴量を抽出したのち、Convolution と Up Sampling によって元の画像サイズにアップサンプリングし最終的に画素ごとに Softmax 関数を用いてどのオブジェクトに属するかを判別する構造を持つ。この構造のうち、ダウンサンプリングを行う部分をエンコーダ（Encoder）、アップサンプリングを行う部分をデコーダ（Decoder）と呼ぶ。

しかし、入力層から最終層まで隣接する層同士の結合のみで構成される線形的なモデル構造では、ダウンサンプリングによって輪郭情報が失われてしまうという問題がある。

そこで輪郭情報を保持するためのモデルとして後述する U-Net や PSPNet のモデル構造が開発されていった。

2.1.1. U-Net

U-Net は、細胞画像のセマンティックセグメンテーションのタスクにおいて開発された CNN のモデル構造である。

特徴的なものとして、デコーダと同じ特徴マップを持つエンコーダの出力をデコーダへの入力とする U 字型の構造を持つことがあげられ、この構造によりダウンサンプリングにより失われる前の輪郭情報を深い層もで保持することが可能となる。U-Net のモデル構造を図 2.1 に、詳細な層構造を表 2.1 に示す。

図 2.1 においてデコーダ D_n の入力はエンコーダ E_n から出力される特徴マップとデコーダ D_{n-1} の出力をアップサンプリングした特徴マップを結合した特徴マップとなる。また、Up Sampling の際には 2×2 の Convolution 層を設置し、エンコーダから結合される特徴マップと同じサイズになるようフィルタ枚数を設定している。

表 2.1 における Layer は層の種類、Output は出力する特徴マップのサイズ、Filters は Convolution のフィルタの枚数、Kernel は Convolution のフィルタのサイズを表している。

また、Dropout 層の Dropout 率は 0.5 に設定されている。

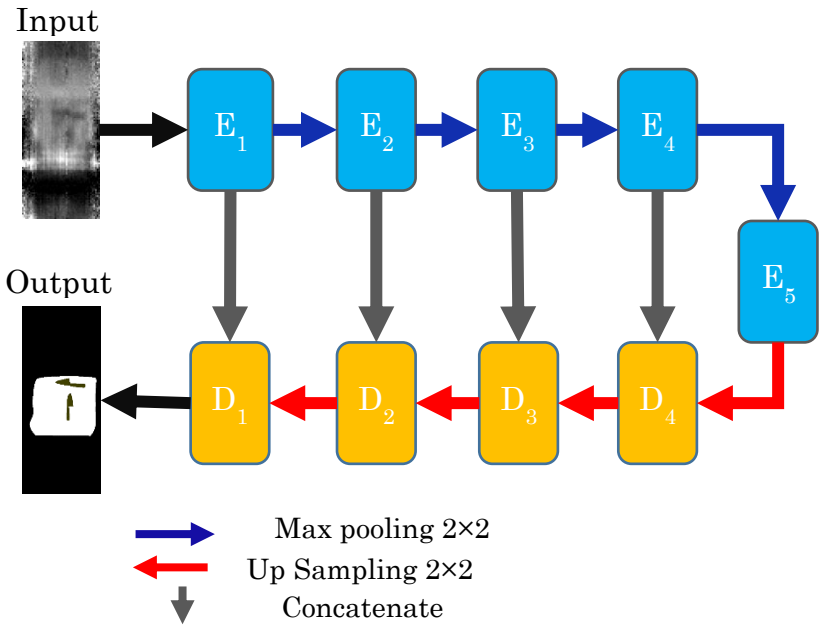

図 2.1　U-Net のモデル構造

表 2.1　U-Net の層構造

Block	Layer	Output	Filters	Kernel
E_1	Convolution	224×224	64	3
	Convolution	224×224	64	3
E_2	Convolution	112×112	128	3
	Convolution	112×112	128	3
E_3	Convolution	56×56	256	3
	Convolution	56×56	256	3
E_4	Convolution	28×28	512	3
	Convolution	28×28	512	3
	Dropout	28×28	512	
E_5	Convolution	14×14	1024	3
	Convolution	14×14	1024	3
	Dropout	14×14	1024	
D_4	Convolution	28×28	512	3
	Convolution	28×28	512	3
D_3	Convolution	56×56	256	3
	Convolution	56×56	256	3
D_2	Convolution	112×112	128	3
	Convolution	112×112	128	3
D_1	Convolution	224×224	64	3
	Convolution	224×224	64	3
	Softmax	224×224	9	1

2.1.2. PSPNet

PSPNet は、風景画像のセマンティックセグメンテーションのタスクにおいて開発された CNN のモデル構造である。

PSPNet の構造の特徴として、Pyramid Pooling モジュールを用いることがあげられる。

Pyramid Pooling モジュールは特徴マップに対して 1、2、3、6 のそれぞれの異なるウインドウサイズで Convolotino を行い最終的に結合する手法であり、様々なサイズで Convolution を行うことで輪郭情報を取得する。

Pyramid Pooling モジュールに用いる特徴マップには ResNet の出力を用いる。

2.1.3. Light-Weight Asymmetric U-Net

Light-Weight Asymmetric U-Net[3]は、ファッション画像のセマンティックセグメンテーションにタスクにおいて U-Net を改良し開発されたモデルである。

U-Net の構造に加えて、デコーダのサイズより 1 段階ダウンサンプリングされたエンコーダの出力する特徴マップにたいしてアップサンプリング処理してものを新たに入力とするスキップコネクションを持つ。また、U-Net のデコーダにおけるフィルタ数をすべて 128 まで減らすことでパラメータ数を削減することでセマンティックセグメンテーションの推論時間を短縮化することに成功している。

モデル構造を図 2.2、詳細な層構造を表 2.2 に示す。

図 2.2 において、デコーダ D_n の入力はエンコーダ E_n から出力される特徴マップ、デコーダ D_{n+1} の出力をアップサンプリングした特徴マップ、さらにエンコーダ E_{n+1} の出力を入力とするスキップコネクション C_n から出力される特徴マップを結合したものとなる。

また、表 2.2 において表 2.1 と同じパラメータに加え、Stride はフィルタのスライド幅、Padding はパディングのサイズ、Activation は層に用いる活性化関数を表している。また、フィルタ枚数の Classes はアノテーション画像に含まれるクラス数となる。

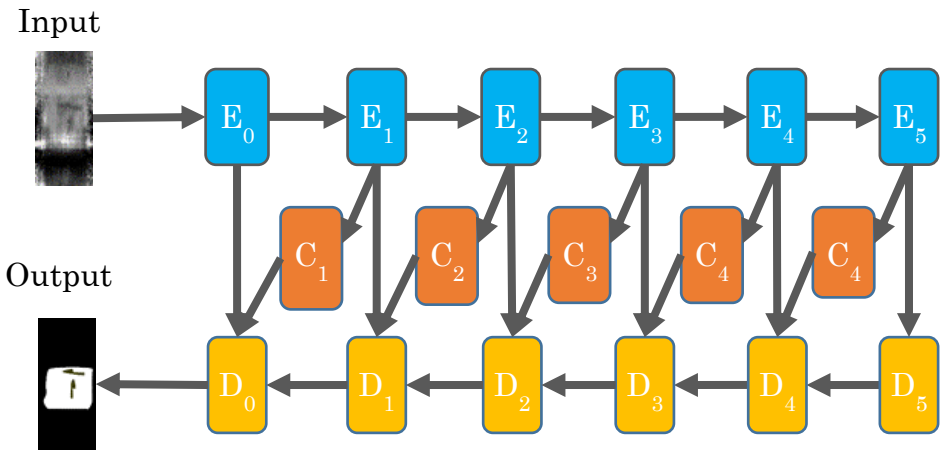

図 2.2　Light-Weight Asymmetric U-Net のモデル構造

表 2.2　Light-Weight Asymmetric U-Net の層構造

Block	Layer	Output	Filters	Size	Stride	Pad	activation
E_0	Convolution	224×224	32	5	1	2	ReLu
	Convolution	224×224	32	3	1	0	ReLu
	Convolution	224×224	64	1	1	0	ReLu
E_1	Convolution	112×112	64	4	2	1	ReLu
	Convolution	112×112	64	3	1	1	ReLu
	Convolution	112×112	128	1	1	0	ReLu
E_2	–”–	56×56	256	–”–	–”–	–”–	–”–
E_3	–”–	28×28	512	–”–	–”–	–”–	–”–
E_4	–”–	14×14	1024	–”–	–”–	–”–	–”–
E_5	–”–	7×7	2048	–”–	–”–	–”–	–”–
D_0	Convolution	224×224	128	3	1	1	ReLu
	Convolution	224×224	Classes	1	1	0	Softmax
D_1	Convolution	112×112	128	3	1	1	ReLu
	Convolution	112×112	128	3	1	1	ReLu
	Up Sampling	224×224	Classes	2	2	0	
D_2	–”–	112×112	–”–	–”–	–”–	–”–	–”–
D_3	–”–	56×56	–”–	–”–	–”–	–”–	–”–
D_4	–”–	28×28	–”–	–”–	–”–	–”–	–”–
D_5	–”–	14×14	–”–	–”–	–”–	–”–	–”–
C_1	Convolution	112×112	32	3	1	1	ReLu
	Up Sampling	224×224	32	2	2	0	
C_2	–”–	112×112	–”–	–”–	–”–	–”–	–”–
C_3	–”–	56×56	–”–	–”–	–”–	–”–	–”–
C_4	–”–	28×28	–”–	–”–	–”–	–”–	–”–
C_5	–”–	14×14	–”–	–”–	–”–	–”–	–”–

2.1.4.　Label-Pooling U-Net

Label-Pooling U-Net[4]は、ファッション画像のセマンティックセグメンテーションのタスクにおいて Light-Asymmetric U-Net にさらに改良を加え開発されたモデルである。

Label-Pooling のモデル構造を図 2.3 に詳細な層構造を表 2.3 に示す。

Label-Pooling U-Net は、Light-Weight Asymmetric U-Net の層構造に変更を加えた上に補助的ロスをさらに付け加えたモデルである。

補助的ロスは IPL(Image Pyramid Pooling Loss), SPL(Segmentation Pyramid Loss), LPL(Label-Pooling Loss)の 3 種類ある、3 種類の補助的ロスと最終ブロックの出力から計算されるロスを合計したものがモデル全体のロスとなる。

まず、IPL は Encoder の 1 番目から 4 番目のブロックの出力において Sigmoide 関数を

適用したのち入力画像をダウンサイズした特徴マップと比較することでロスを計算し、各ブロックごとの Loss の平均を取る。SPL は Decoder の出力に近いブロックの 1 番目から 4 番目の 4 個のブロックの出力において Softmax 関数を適用、アノテーション画像をダウンサイズした特徴マップと比較することでロスを計算し、それぞれの Loss の平均を取る。

LPL は Decoder のすべてのブロックの出力において Sigmoid 関数を適用、アノテーション画像を Label Pooling した特徴マップと比較することでロスを計算し、各ブロックごとの Loss の平均を取る。

Label-Pooling は、補助的ロスの LPL の計算に用いるため開発された Pooling の手法であり、出力画像を one-hot ベクトル化した特徴マップについて Max Pooling の手法を行っていく手法である。セマンティックセグメンテーションにおける one-hot ベクトルは、それぞれの画素において分類される対象物体のクラスを 1、分類されない対象物体のクラスを 0 として表す。そのため、one-hot ベクトル化した特徴マップを Max Pooling していくことにより、通常のダウンサイズ手法では失われてしまう画素のクラス情報を保持しつつのダウンサイズが可能となる。

B1-B4 のブロックについては入力をフィルタ数 64 の 3×3 Convlution 層で畳み込んだのちに Sigmoid 関数を適用した特徴マップを出力し、L0-L4 のブロックについては入力をフィルタ数 128 の 3×3 Convlution 層で畳み込んだのちに Softmax 関数を適用した特徴マップを出力する。

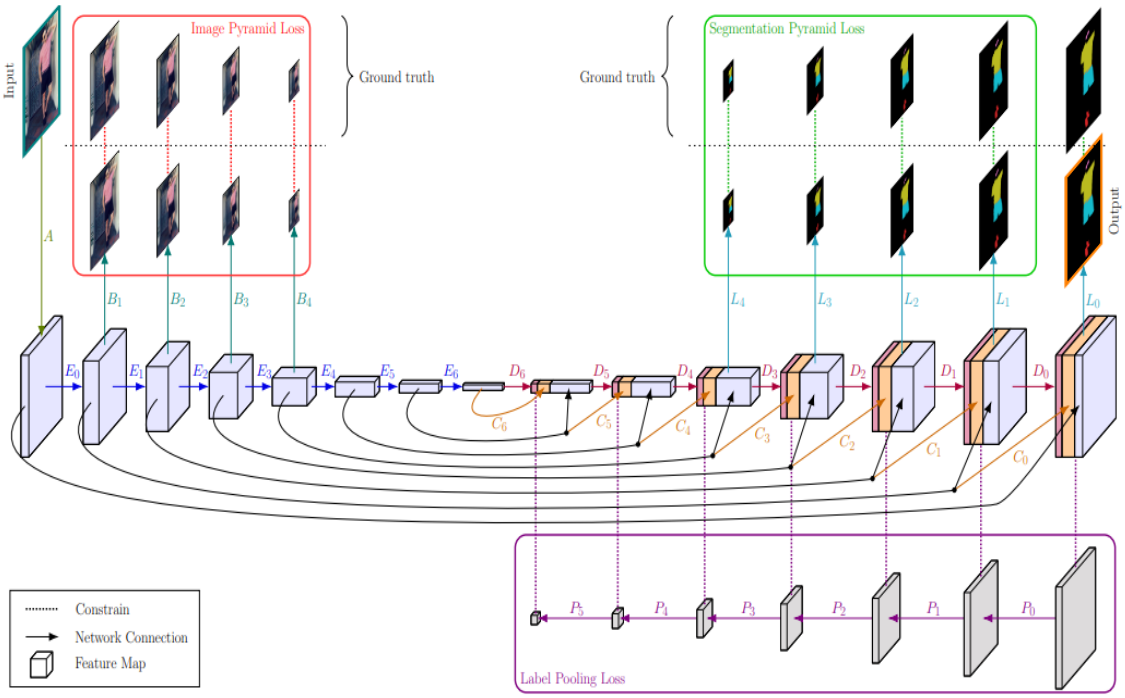

図 2.3　Label-Pooling U-Net のモデル構造(文献[4]より引用)

表 2.3　Label-Pooling U-Net の層構造

Block	Layer	Output	Filters	Size	Stride	Pad	activation
A	Convolution	224×224	32	5	1	2	ReLu
	Convolution	224×224	32	3	1	0	ReLu
	Convolution	224×224	64	1	1	0	ReLu
E_0	Convolution	112×112	64	4	2	1	ReLu
	Convolution	112×112	128	3	1	1	ReLu
	Convolution	112×112	64	1	1	0	ReLu
E_1	–”–	56×56	256	–”–	–”–	–”–	–”–
E_2	–”–	28×28	512	–”–	–”–	–”–	–”–
E_3	–”–	14×14	1024	–”–	–”–	–”–	–”–
E_4	–”–	7×7	2048	–”–	–”–	–”–	–”–
E_5	Convolution	3×3	1024	3	3	1	ReLu
	Convolution	3×3	2048	3	1	1	ReLu
	Convolution	3×3	1024	1	1	0	ReLu
E_6	Convolution	1×1	1024	3	1	1	ReLu
	Convolution	1×1	2048	3	1	1	ReLu
	Convolution	1×1	1024	1	1	0	ReLu
D_0	Convolution	112×112	128	3	1	1	ReLu
	Convolution	112×112	Classes	3	1	1	Sigmoid
	Up Sampling	224×224	Classes	2	2	0	
D_1	–”–	112×112	–”–	–”–	–”–	–”–	–”–
D_2	–”–	56×56	–”–	–”–	–”–	–”–	–”–
D_3	–”–	28×28	–”–	–”–	–”–	–”–	–”–
D_4	–”–	14×14	–”–	–”–	–”–	–”–	–”–
D_5	Convolution	3×3	128	3	1	1	ReLu
	Convolution	3×3	Classes	3	1	1	Sigmoid
	Up Sampling	7×7	Classes	3	3	0	
D_6	Convolution	1×1	128	3	1	1	ReLu
	Convolution	1×1	Classes	3	1	1	Sigmoid
	Up Sampling	3×3	Classes	3	1	0	
C_0	Convolution	112×112	128	3	1	1	ReLu
	Up Sampling	224×224	128	2	2	2	
C_1	–”–	112×112	–”–	–”–	–”–	–”–	–”–
C_2	–”–	56×56	–”–	–”–	–”–	–”–	–”–
C_3	–”–	28×28	–”–	–”–	–”–	–”–	–”–
C_4	–”–	14×14	–”–	–”–	–”–	–”–	–”–
C_5	Convolution	3×3	128	3	1	1	ReLu
	Up Sampling	7×7	128	3	3	1	
C_6	Convolution	1×1	128	3	1	1	ReLu
	Up Sampling	3×3	128	3	3	0	

2.2. オブジェクト認識

2.2.1. ResNet

ResNet[5]は、2015 年に考案された CNN のモデルの案である。

ResNet の大きな特徴として、Residual Block を開発したことにある。

Residual Block は、ある Convolution 層の出力において Skip Connection を用いて入力との残差を取るブロック構造である。

通常 CNN では、層を深くすることにより複雑で大きなパターンを学習することが可能であると考えられている。しかし、層を深くしすぎると、情報の伝達を妨げる表現のボトルネックや誤差逆伝播がうまく行かなくなる勾配消失問題が発生し性能が悪化してしまう問題があった。

そこで、ある層における出力だけでなく入力との残差を次の層への入力とする Residual Block を導入することで情報の損失に対処し、層の深さを押し上げることに成功した。

2.2.2. DenseNet

DenseNet[6]は、オブジェクト認識において ResNet にさらに改良を加えたモデルである。

ResNet では、Convolution の入力と出力の残差を取る Residual Block を用いるが、DenseNet では Convolution の入力と出力を直接結合する DB（Dense Block）を用いる。Convolution の層をすべて直接に結合するため、特徴の伝達を Residual Block よりさらに強化することで Convolution のフィルタ枚数を削減することでモデル全体のパラメータ削減と特徴の効率的な利用が可能となった。

DenseNet をセマンティックセグメンテーションのために改良したものが FC-DenseNet[7]となる。

FC-DenseNet は、ダウンサンプリングを行うエンコーダ部分を DB と Transition Down、アップサンプリングを行うデコーダ部分を DB と Trasition UP で構成される。さらにエンコーダ部分の DB に出力する特徴マップについて、デコーダ部分と同じサイズを持つ特徴マップへ結合するという U-Net に由来した構造を持つことも特徴である。また、ダウンサンプリング前の入力画像に対して通常のウィンドウサイズが 3×3 である Convolution 層が設置されそのフィルタ枚数を NB Filter で表され、DB 内の 1 個あたりの Convolution 層のファイル他枚数は Growth Rate として与えられる。

Transition Down にはストライド幅が 2 である 3×3 の Convolution 層が用いられそのフィルタ枚数は、直前の DB 内に含まれる Convolution 層のフィルタ枚数の合計である。また、Transition Up にはストライド幅が 2 である 3×3 の Deconvolution 層が用いられそのフィルタ枚数は、直前の DB 内に含まれる Convolution 層のフィルタ枚数の合計である。

図 2.4 は文献[7]から引用した FC-DenseNet の概要図である。

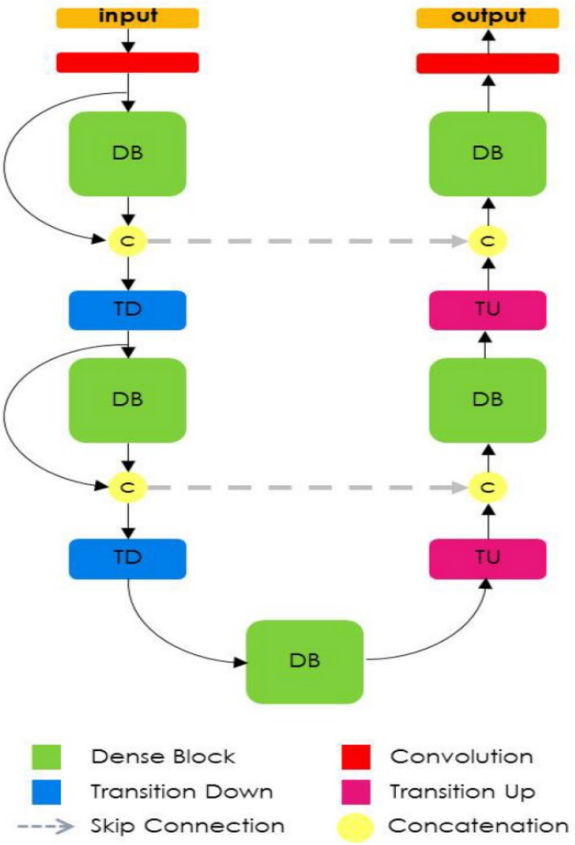

図 2.4　FC-DenseNet の概要図(文献[7]より引用)

2.3.　データオーギュメンテーション

データオーギュメンテーションは、画像認識における機械学習において画像に人為的に反転、切り取り、回転、拡大収縮等の編集処理を加えることに学習するデータ量を量増しする手法である。データオーギュメンテーションを行う処理の種類を増やすことにより、学習するデータ量も比例して増やすことが可能であるうえ、本来学習データに含まれないために起きる対象物体の位置や角度に違いによる誤認識を減らすことも期待できる。しかし、有効でないデータオーギュメンテーションを行うと精度が向上しないうえにデータ数の増加による学習時間のみが増大してしまうため、データセットに合わせたデータオーギュメンテーションの選択が必要となる。

また、セマンティックセグメンテーションのタスクにおいて正解ラベルは画像になるため、入力画像とアノテーション画像に同じオーギュメント処理を加える必要がある。

2.3.1.　Mixup

Mixup[8]は、2017 年に開発されらデータオーギュメンテーションの手法である。

Mixup の大きな特徴として、学習画像を 2 枚混ぜ合わせることで新しい学習データを生成する手法である。

混ぜ合わせる手法として、まずパラメータ α に設定される β 分布によってランダムに λ を 0 から 1 の範囲で生成する。そして、x を入力するモデルデータ、y を正解ラベルを One-hot ベクトル化したものとしたとき、2 種類のデータの組（x_i、y_i）、（x_j、y_j）に対しそれぞれ重み λ、1-λ を与え足し合わせることで合成される 1 つのデータの組を生成していく。

このオーギュメンテーションにおいて、ハイパーパラメータは α であり任意に値が設定される。

$$\begin{gathered}\tilde{x} = \lambda x_i + (1-\lambda)x_j \\ \tilde{y} = \lambda y_i + (1-\lambda) \\ \lambda \in [0,1], \quad \lambda \sim \beta(\alpha,\alpha), \qquad \alpha \in (0,\infty) \\ x\text{は画像データ、} \quad y\text{は正解ラベル}(\text{one} - \text{hot ベクトル})\end{gathered} \tag{2.1}$$

第3章　セマンティックセグメンテーションの精度比較

3.1.　比較の条件

既存の CNN モデルである U-Net、Light-Weight Asymmetric U-Net、 Label-Pooling U-Net、PSPNet の 4 種類で不審物検知の精度比較を行う。

本研究で、比較に用いる画像は W 帯を用いるパッシブイメージング技術により得られる画像で図 3.1 のように不審物をプラスチック板に張り付け布をかけた状態の様子を W 帯によって微弱な電波を計測覆われている物体の透過率を画像化することでパッシブイメージャー画像を得ることができる。

不審物は刃物、銃、携帯、液体、粉体、模擬爆弾の 6 種類であり、アノテーション画像に含まれる領域は不審物 6 種類に背景領域と人物領域を加えた 9 種類である。

また、図 3.1 より不審物をプラスチック板に張り付けて撮影を行っているが、本研究ではこのプラスチック板に当たる部分を人物領域と仮定してセメンティック・セグメンテーションのアノテーション画像が作成されている。パッシブイメージャ画像はすべてで 1,008 枚であり、このうち 882 枚を学習用とし、126 枚は検証用として用い、モデルに入力する際には、224×224 のサイズにリサイズ処理を行う。

実験環境では、Keras のライブラリを用いてモデルの作成および学習・評価プログラムの作成を行う。ネットワークの学習に用いる GPU は GeForce GTX 1080 を用いる。

続いて学習条件については、バッチサイズは 8 で、イテレーション数は 882 である。エポック数は 60 で検証を行った。損失関数は Categorical_CrossEntropy、最適化関数は Adam を用い、学習率は 10^{-4} とする。PSPNet に用いる ResNet の種類には、ResNet50 を用いる。

また、データオーギュメンテーションとしては一般的に用いられる水平反転と Scale Augementation を適用する。水平反転は、画像を水平方向に反転させることで画像を量増しする手法であり、本研究ではデータの抽出ごとにランダムに画像を反転し、その確率を 0.5 とする。また、Scale Augmentation は元の画像からランダムな箇所を切り抜くことで画像を量増しする手法であり、本研究では、入力する画像を 224×224 にリサイズした後に 1-1.5 倍の範囲でランダムに拡大したのち、モデルに入力するサイズである 224×224 のウィンドウでランダムな箇所を切り抜くことで Scale Augmentation を実現する。

水平反転と Scale Augmentation をアノテーション画像のセグメンテーションマップに適用した例を図 3.2、図 3.3 に示す。

また、実験環境、学習条件を簡易的にまとめたものを表 3.1 に、データオーギュメンテーションの条件をまとめたものを表 3.2 に示す。

モデルの精度評価の手法については IoU 値を用いる。

IoU とは、Interscetion over Union の略語でありセマンティックセグメンテーションの評価指標として用いられる。IoU は予測領域と正解領域の共通領域となる面積に対して予測領域と正解領域の和集合領域の面積で除算して表される評価手法である。あるクラスに

対する分類結果において、モデルの予測が陽性で実際にそのクラスに分類される画素数を TP（True Positive)、モデルの予測が陽性だが実際はそのクラスに分類されない画素数を FP（False Positive)、モデルの予測が陰性だが実際はそのクラスに分類される画素数を FN（False Positive）とすると、IoU 値は次のような式で表される。

$$IoU = \frac{TP}{\text{予測領域} \cap \text{正解領域}} = \frac{TP}{TP + FP + FN} \tag{3.1}$$

また、全てのクラスにおける IoU 値の平均を取った値を mIoU と表す。

セマンティックセグメンテーションにおける TP、FP、FN を図で表したものを図 3.4 に示す。

表 3.1　実験環境および学習条件

実験環境:	
GPU	GeForce GTX 1080
フレームワーク	Keras
データセット	パッシブイメージャ画像
学習用枚数	882 枚
検証用枚数	126 枚
アノテーション画像のクラス	背景、人物、刃物、銃、携帯電話、液体、粉体、模型爆弾、ファントム
学習条件:	
入力サイズ	224×224
バッチサイズ	8
イテレーション数	882
エポック数	60
最適化関数	Adam
損失関数	Categorical_CrossEntropy
学習率	10^{-4}
PSPNet に用いる ResNet	ResNet50

表 3.2　データオーギュメンテーションの条件

水平反転:	
反転の確率	0.5
Scale Augmentation:	
拡大範囲	1-1.5 倍
切り取るサイズ	224×224

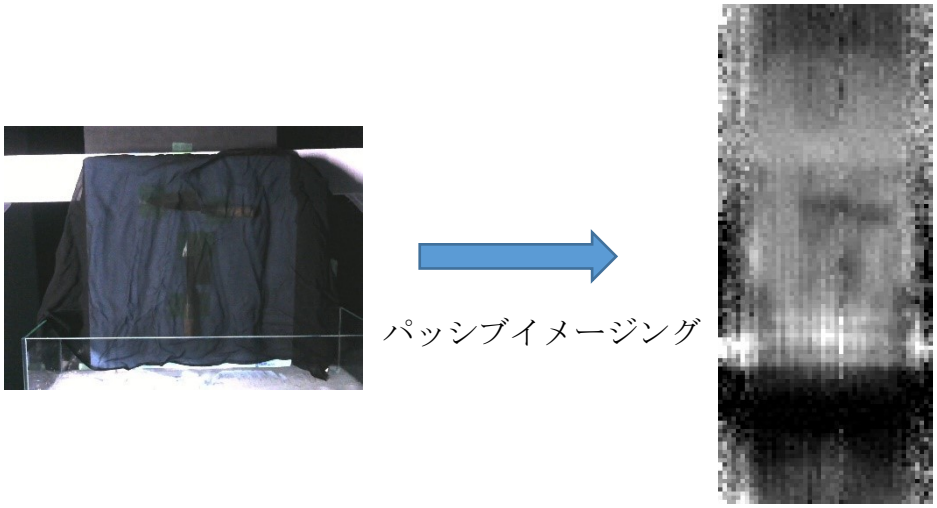

図 3.1　パッシブイメージング技術

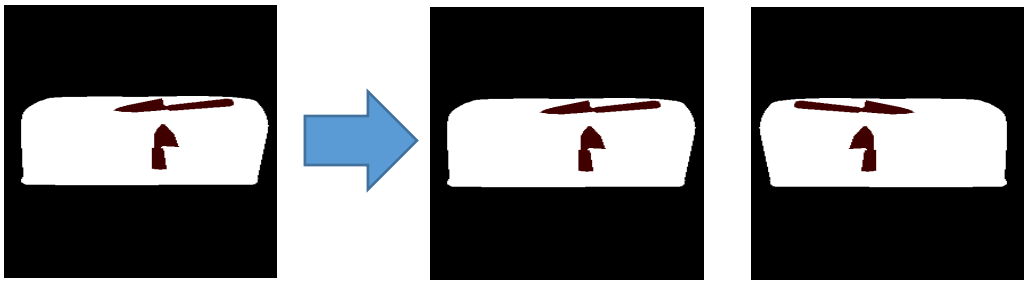

図 3.2　水平反転によるデータオーギュメンテーションの例

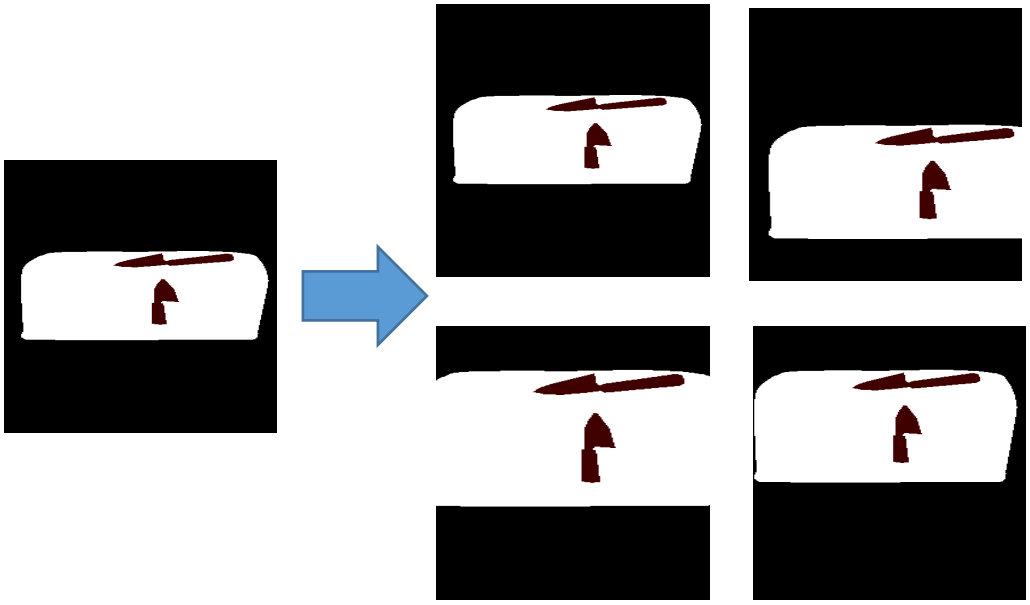

図 3.3　Scale Augmentation によるデータオーギュメンテーションの例

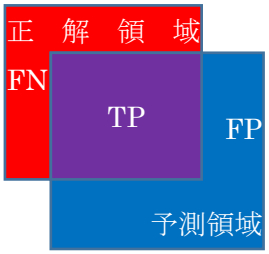

図 3.4　セマンティックセグメンテーションにおける TP、FP、FN

3.2.　比較の結果及び考察

4 種類のモデルについて表 3.1、表 3.2 の条件において学習を行い、検証用データにおいて IoU 値を算出し比較した数値を表 3.3 に示す。また、モデルによって出力される画像の比較を図 3.5 に示す。

表 3.3 において、mIoU は PSPNet が最も高く、クラスごとの比較においても PSPNet が最も高い IoU 値を持つクラスが 6 個ある。

また、刃物領域において他のクラスに比べて IoU 値が低くなる原因として、他のクラスと比べ領域の面積が狭く形が複雑なため、予測領域と正解領域の誤差が生じやすいことが原因であると考えられる。

図 3.5 において、出力画像の比較を行ってみると特に刃物の領域の部分で PSPNet が IoU 値が高いにも関わらず U-Net による予測の方が外形を再現できている様子が見受けられる。

そのため、PSPNet は正解領域の位置予測に優れている反面、細かい輪郭情報を予測する

面においてはU-Netのモデル構造の方が優れていると考えられる。

表 3.3　IoU 値の比較

Class	U-Net[1]	LWA U-Net[3]	LP U-Net[4]	PSPNet[2]
背景	0.979	**0.981**	0.980	**0.981**
人物	0.869	0.870	0.867	**0.875**
刃物	0.536	**0.547**	0.537	**0.547**
銃	0.729	0.722	0.722	**0.732**
携帯	**0.756**	0.721	0.754	0.739
液体	0.688	0.791	0.897	**0.924**
粉体	0.714	0.746	0.681	**0.851**
模擬爆弾	0.785	**0.815**	0.779	0.806
ファントム	**0.929**	0.863	0.879	0.836
mIoU	0.776	0.784	0.788	**0.810**

LWA U-Net: Light-Weight Asymmetric U-Net
LP U-Net: Label-Pooling U-Net

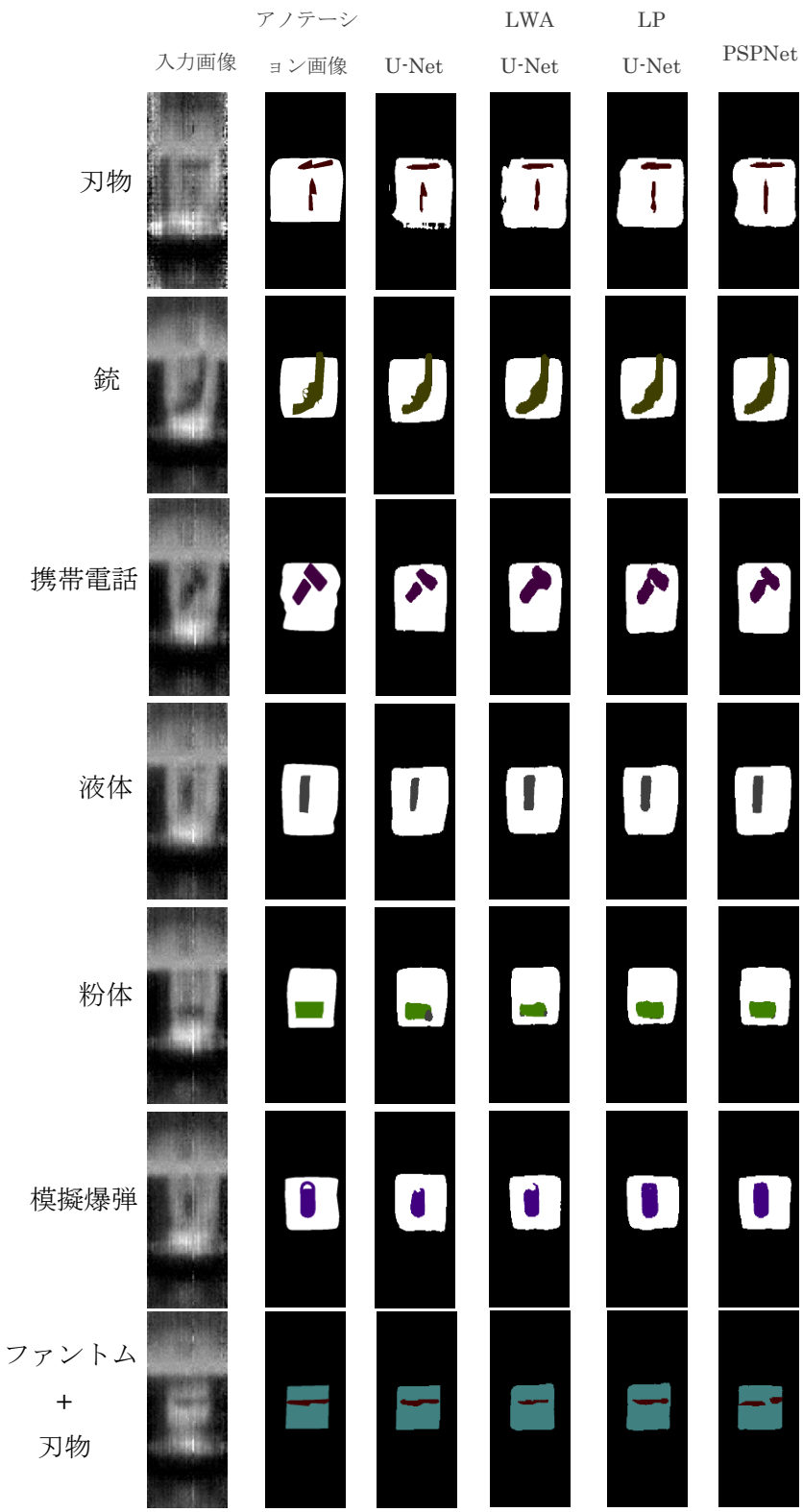
入力画像
アノテーション画像
U-Net
LWA U-Net
LP U-Net
PSPNet
刃物
銃
携帯電話
液体
粉体
模擬爆弾
ファントム＋刃物

図 3.5　出力画像の比較

第4章　Mixup の適用及びオーギュメンテーションの違いによる精度の比較

4.1.　実験条件

Mixup のデータオーギュメンテーションを用いて学習を行い、一般的に用いられる水平反転と Scale Augmentation を用いたオーギュメンテーションと比較する。

また、比較に用いるモデルとしては U-Net、 Light-Weight Asymmetric U-Net、 Label-Pooling U-Net、PSPNet を用いる。

さらに Mixup のパラメータ α の値は 0.2 に設定する。また、数式 2.1 における（x_i、y_i）のデータの組は学習データから順番に、（x_j、y_j）のデータの組は学習データからランダムに抽出する。重み λ は、データの抽出ごとにランダムに決定する。これらの Mixup の条件をまとめたものを表 4.1 に示す。また、α の値を 0.2 にしたときの β 分布の様子を図 4.1 に、λ を 0.4 に設定されたときのアノテーション画像のセグメンテーションマップの Mixup の例を図 4.2 に示す。

実験環境と学習条件は表 3.1、Mixup を用いない場合のオーギュメンテーションの条件は表 3.2 と同じものを用いる。

表 4.1　Miup の条件

β 分布の α 値	0.2
データの組（x_i、y_i）	学習データから順番に抽出
データの組（x_j、y_j）	学習データからランダムに抽出
重み λ	データ組の抽出ごとにランダムに決定

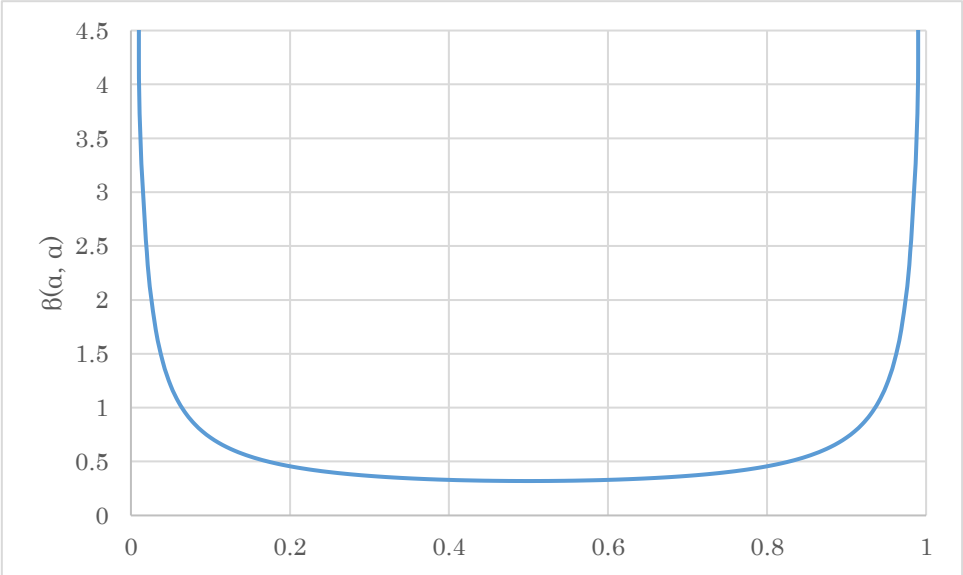

図 4.1　α=0.2 としたときの β 分布の取る値

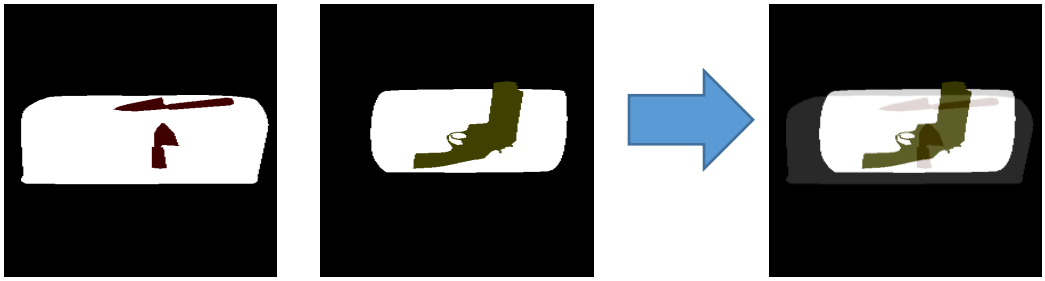

図 4.2　λ=0.4 としたときの Mixup の例

4.2.　比較の結果及び考察

4 種類のモデルにおいて、一般的なオーギュメンテーショと Mixup の 2 種類オーギュメンテーショでそれぞれ学習を行ったのち検証用のデータで IoU 値を算出し、モデルごとおよびオーギュメンテーショごとで IoU 値の比較を示したものを表 4.2 に、Mixup を用いて学習を行ったモデルの出力画像を比較した様子を図 4.3 に示す。

表 4.2 において、データオーギュメンテーション(a)が水平反転と Scale Augmentation(b)を組み合わせた一般的なデータオーギュメンテーション、データオーギュメンテーション（b）が Mixup を表している。

表 4.2 より、Light-Weight Asymmetric U-Net 以外のモデルにおいて Mixup を用いる場合に mIoU の値が向上していて、とくに U-Net では 0.84 の向上がみられる。よって、Mixup オーギュメンテーショはパッシブイメージャー画像を用いた不審物検知に有効であると考えられる。また、U-Net に対して Light-Weight Asymmetric U-Net や Label Pooling U-Net で IoU の大きな向上が見られない理由としては、これらのモデルがファッション画像のセマンティックセグメンテーションに特化したモデル構造に改良したためと考えられる。

また、U-Net に Mixup を適用した場合において mIoU の値が最も高く、また粉体を除く不審物およびファントムのクラスにおいても最も高い値を示す。

さらに、図 4.3 により出力画像を比較すると、U-Net の出力画像は他のモデルと比較してセグメンテーションマップの再現率が高く、とくに銃および模型爆弾のクラスにおいて他のモデルより不審物の外形の再現率が高い。そのため、Mixup を用いる際には、U-Net のモデルが最も高い精度が得られる。

また、表 4.より PSPNet においても Mixup を適用すると液体のクラス以外で IoU 値の向上が見られるが、図 4.3 より外形の再現度は高くないことから細かい輪郭情報の学習が U-Net より劣っているため U-Net ほどの効果は得られなかったのではないかと考えられる。

表 4.2　Mixup を用いる場合の IoU 値の比較

	U-Net[1]		LWA U-Net[3]		LP U-Net[4]		PSPNet[2]	
Class	(a)	(b)	(a)	(b)	(a)	(b)	(a)	(b)
背景	0.979	0.980	0.981	0.980	0.980	0.979	**0.981**	**0.981**
人物	0.869	0.875	0.870	0.870	0.867	0.860	0.875	**0.881**
刃物	0.536	**0.598**	0.547	0.511	0.537	0.516	0.547	0.557
銃	0.729	**0.783**	0.722	0.728	0.722	0.720	0.732	0.745
携帯	0.756	**0.825**	0.721	0.711	0.754	0.780	0.739	0.800
液体	0.688	**0.941**	0.791	0.725	0.897	0.909	0.924	0.897
粉体	0.714	0.815	0.746	0.718	0.681	0.817	0.851	**0.867**
模擬爆弾	0.785	**0.948**	0.815	0.838	0.779	0.815	0.806	0.860
ファントム	0.929	**0.974**	0.863	0.877	0.879	0.912	0.836	0.886
mIoU	0.776	**0.860**	0.784	0.773	0.788	0.812	0.810	0.830

(a)：水平反転＋Scale Augmentation,　(b)：Mixup

LWA U-Net：Light-Weight Asymmetric U-Net

LP U-Net：Label Pooling U-Net

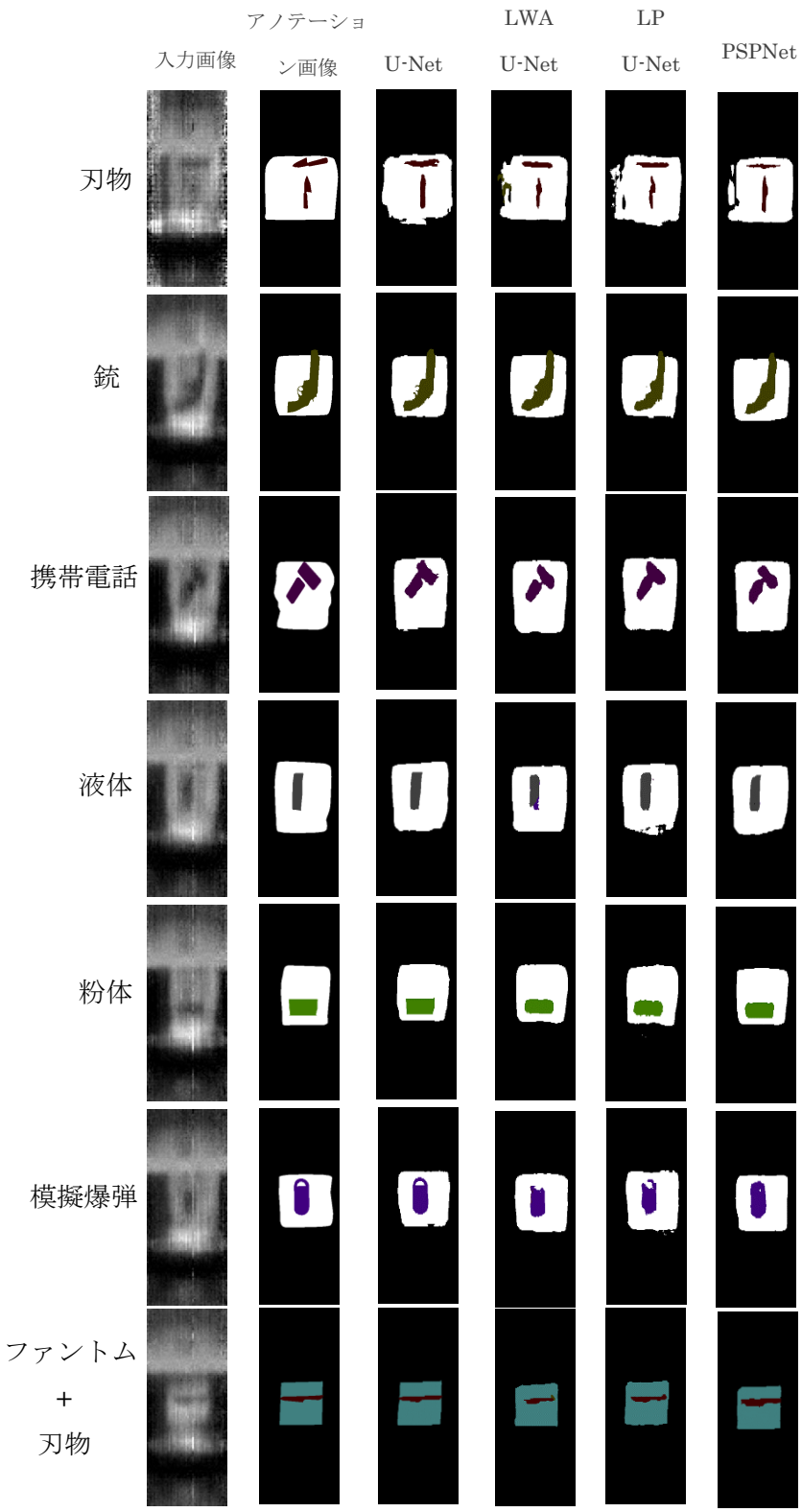

図 4.3　Mixup を用いる場合の出力画像の比較

第5章　U-Net のモデル改良及び改良モデルの違いによる精度の比較

5.1.　Residual Block を導入したモデル改良

4 種類の既存の CNN モデルで Mixup を適用して IoU 値を比較したところ、U-Net が最も高い mIoU を計測したため、U-Net に改良を加えること精度が向上ができないかを検討する。

本研究では、U-Net のエンコーダ（Encoder）に Residual Block を設置するモデル、デコーダ（Decoder）に Residual Block を設置するモデル、モデル全体に Residual Block を設置するモデルを用意し検証を行った。3 種類のモデル構造をそれぞれ図 5.1、図 5.2、図 5.3 に示す。

Residual Block を実現するために、エンコーダまたはデコーダのブロックの入力を Skip Connection でブロックの出力と同じ特徴マップのサイズに変形する処理が必要となる。

そのため、エンコーダに用いる Skip Connection には 1×1 の Convolution 層を設置し、そのフィルタ枚数はブロック内の Convolution 層のフィルタ枚数同数になるように設定する。デコーダに用いる Skip Connection の場合、Skip Connection への入力をエンコーダと結合される直前の特徴マップを用いるため、デコーダのブロックから出力される特徴マップのサイズと等しいことから Convolution 層は設置しない。

3 種類のモデルとも表 3.1 と同じ実験環境および学習条件で学習を行う。また、データオーギュメンテーションは Mixup をもちいて表 4.1 と同じ条件で学習データの量増しを行う。

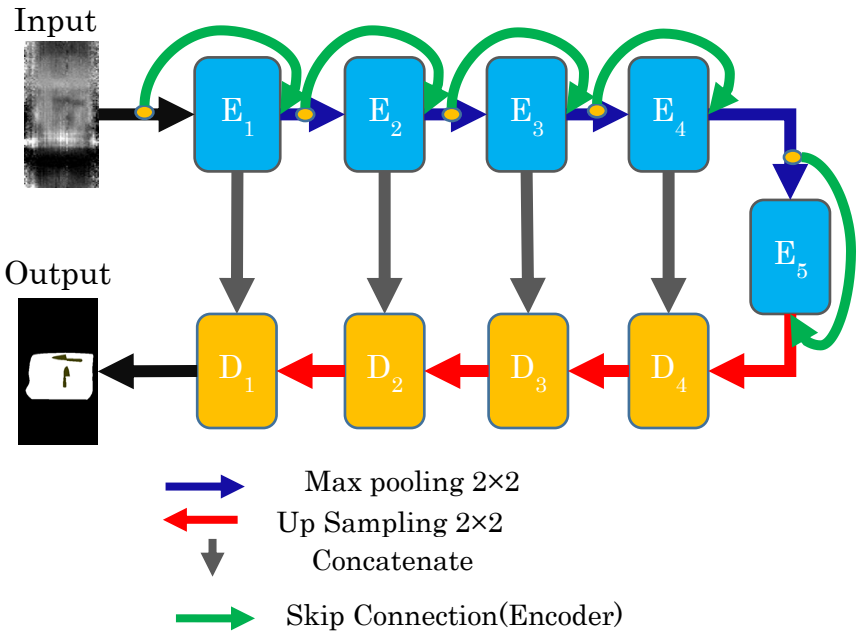

図 5.1　エンコーダに Residual Block を設置する U-Net のモデル構造

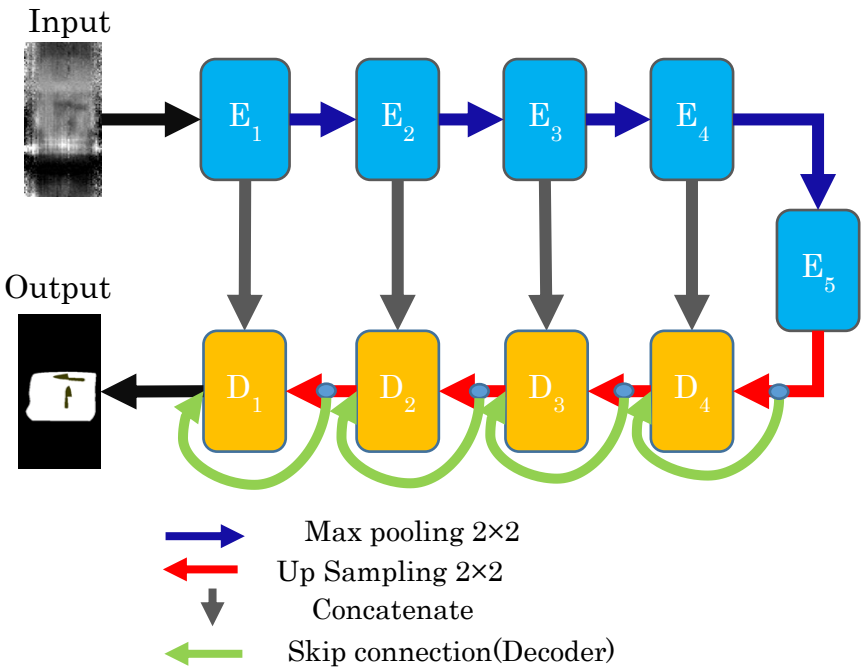

図 5.2　デコーダに Residual Block を設置する U-Net のモデル構造

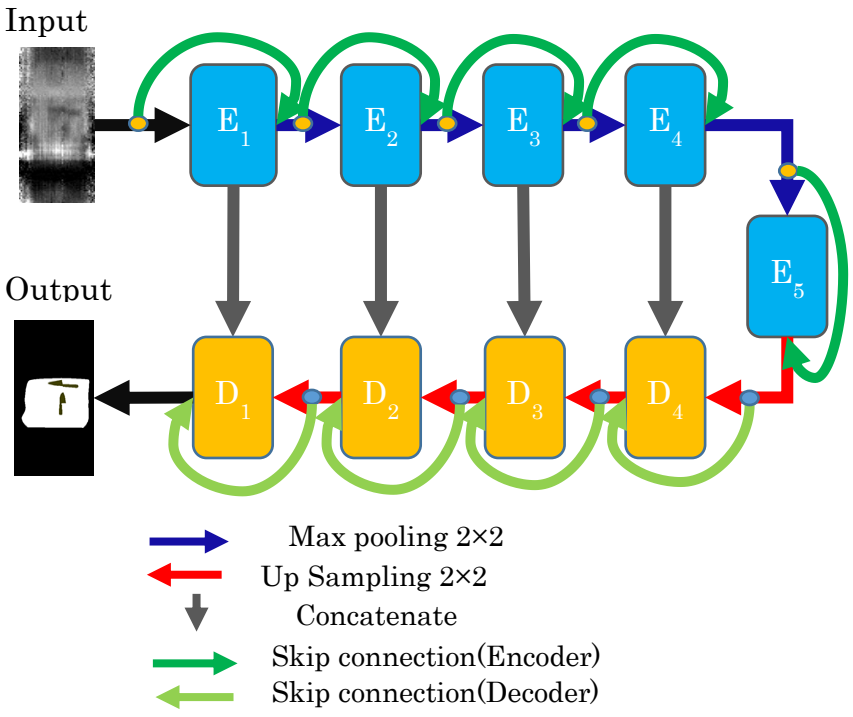

図 5.3　モデル全体に Residual Block を設置する U-Net のモデル構造

5.2. Dense Block を導入したモデル改良

U-Net に Dense Block を導入することで、精度が向上ができないかを検討する。

本研究では、U-Net のエンコーダ（Encoder）を Dense Block に置き換えるモデル、デコーダ（Decoder）を Dense Block に置き換えるモデル、モデル全体が Dense Block となる FC-Densenet の 3 種類のモデルで検証を行った。3 種類のモデル構造をそれぞれ図 5.4、図 5.5、図 5.6 に示し、Dense Block のパラメータおよび学習条件それぞれを表 5.1、表 5.2、表 5.3 に示す。

Dense Block をエンコーダに置き換える際は、Max Pooling を Trainsition Down に置き換え、Dense Block をデコーダに置き換える際は、Up Sampling を Transition UP に置き換える。また、表 5.1、表 5.2、表 5.3 において NB-Filter はエンコーダを DB に置き換える際に入力と Dense Block の間に設置する 3×3Convolution 層のフィルタ枚数、Growth Rate は Dense Block 内の 1 個あたりの 3×3Convolution 層のフィルタ枚数、Layers_per_Block は、Dense Block 内の Convolution 層の数を入力に近い Block から順に示したパラメータとなる。

3 種類のモデルともバッチサイズ、イテレーション数、エポック数以外の条件については表 3.1 と同じ実験環境および学習条件で学習を行う。Dense Block をエンコーダに置き換える場合と Dense Block をデコーダに置き換える場合はエポック数 60 では学習が収束しないためエポック数を 100 に設定する。また、FC-DenseNet を学習させる際にバッチサイズ 8 に設定すると学習の際のメモリ使用量が GPU のメモリ量を超えてしまいかつ学習の収束が遅いため、バッチサイズを 4、イテレーション数を 1764、エポック数を 200 に設定する。

また、データオーギュメンテーションは Mixup をもちいて表 4.1 と同じ条件で学習データの量増しを行う。

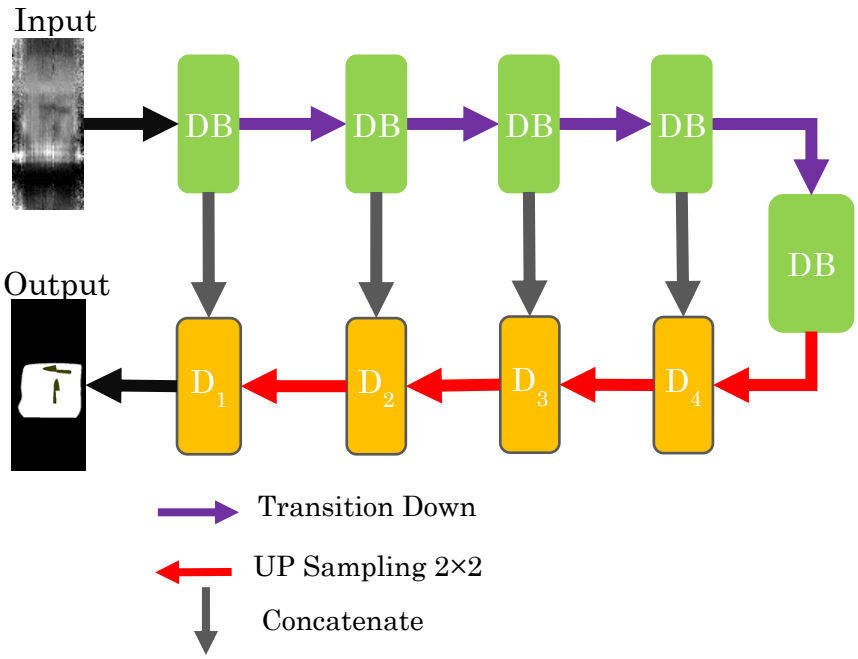

図 5.4　エンコーダを Dense Block に置き換える U-Net のモデル構造

表 5.1　エンコーダを Dense Block に置き換える U-Net のパラメータ

Dense Block:	
NB-Filter	48
Growth Rate	16
Layers-per-Block	[4, 5, 7, 10, 12]
学習条件:	
バッチサイズ	8
イテレーション数	882
エポック数	100

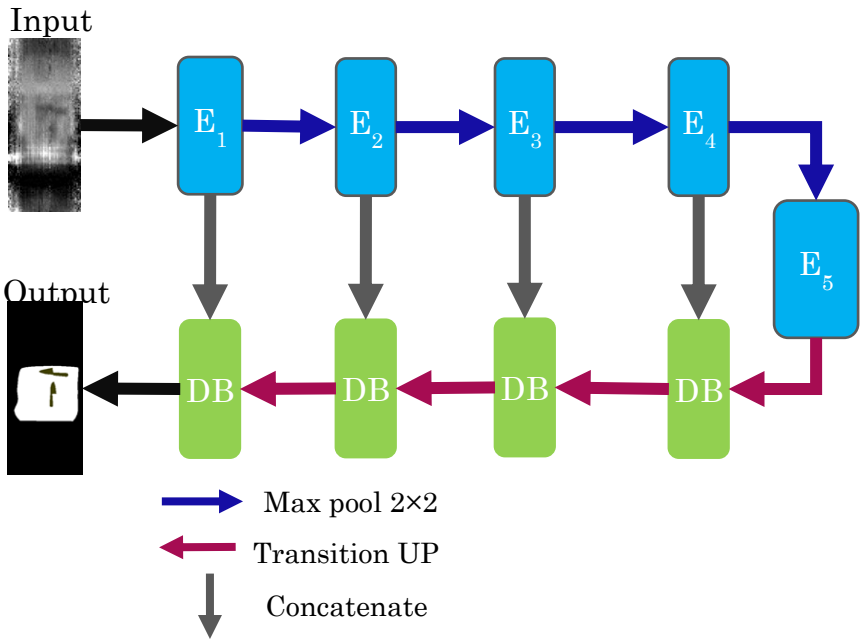

図 5.5　デコーダを Dense Block に置き換える U-Net のモデル構造

表 5.2　デコーダを Dense Block に置き換える U-Net のパラメータ

Dense Block:	
Growth Rate	16
Layers-per-Block	［10, 7, 5, 4］
学習条件:	
バッチサイズ	8
イテレーション数	882
エポック数	100

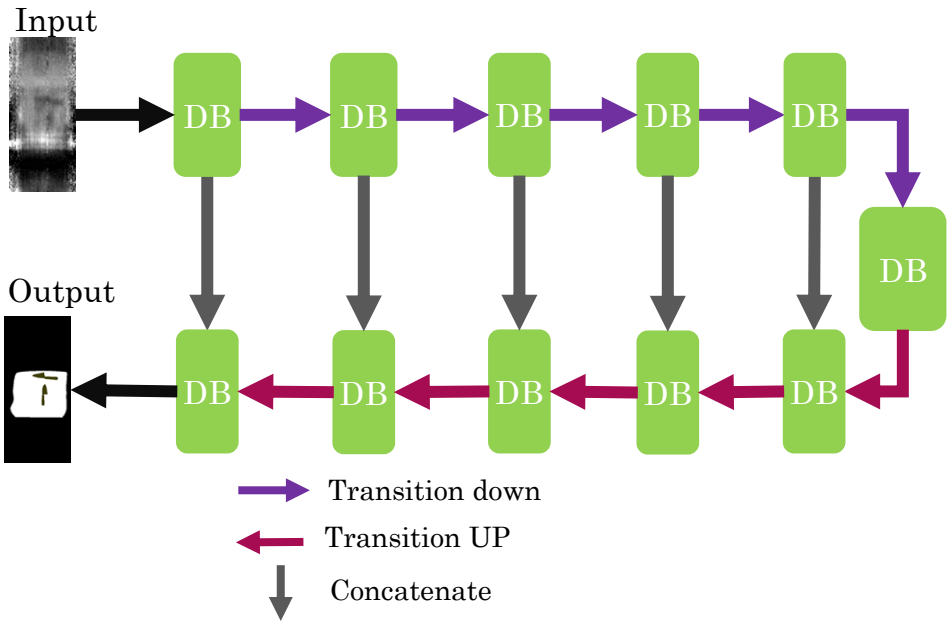

図 5.6　FC-DenseNet のモデル構造

表 5.3　FC-DenseNet のパラメータ

Dense Block:	
NB-Filter	48
Growth Rate	16
Layers-per-Block（Encoder）	［4, 5, 7, 10, 12, 15］
Layers-per-Block（Decoder）	［12, 10, 7, 5, 4］
学習条件:	
バッチサイズ	4
イテレーション数	1764
エポック数	200

5.3. 比較の結果及び考察

従来の U-Net のモデルおよび図 5.1－図 5.6 の改良モデルの計 7 個のモデルについて、モデルの学習を行い、検証用データにおいて IoU 値を計測し比較したものを表 5.4 に、出力画像の比較を行った様子を図 5.7 に示す。

表 5.4 より Residula Block と Decoder Block ともにデコーダのみに適用したものが最も高いという結果になった。よって、Residual Block と Dense Block ともにデコーダーに設置することで入力画像のノイズを影響を小さく効果が期待できると言える。

また、エンコーダに Residual Block および Dense 適用した場合に有効に働かない理由として、入力画像のノイズが大きいため、エンコーダに適用した場合に浅い層のノイズの大きい特徴マップが深い層まで影響を与えてしまうからだと考えられる。

すべての層に Residual Block を用いる場合に未学習が起き、全くモデルが検知を行えないという状態になった。未学習がおきてしまった原因として学習枚数に対しパラメータが大きすぎるという理由が考えられる。

図 5.7 より、モデル全体に Residual Block を設置したモデル、エンコーダに Dense Block を設置したモデル FC-DenseNet の出力画像が他のモデルよりセグメンテーション・マップ再現度の低いことから IoU 値の比較同様、検知の精度が低いことを改めて確認できる。

表 5.4　改良モデルごとの IoU 値の比較

class	U-Net[1]	RB（Residual Block）			DB（Dense Block）		FC-DenseNet[7]
		Encoder	Decoder	All	Encoder	Decoder	
背景	0.980	0.979	0.980	0.820	0.910	**0.982**	0.970
人物	0.875	0.871	0.879	0.0	0.695	**0.883**	0.828
刃物	0.598	0.584	0.600	0.0	0.412	**0.601**	0.439
銃	**0.783**	0.779	0.779	0.0	0.528	0.781	0.565
携帯	0.825	0.792	**0.846**	0.0	0.494	0.808	0.495
液体	0.941	**0.970**	0.911	0.0	0.842	0.914	0.852
粉体	0.815	0.787	0.889	0.0	0.611	**0.919**	0.533
模擬爆弾	0.948	0.938	**0.956**	0.0	0.739	0.882	0.812
ファントム	0.974	0.972	**0.986**	0.0	0.634	0.978	0.863
mIoU	0.860	0.852	**0.869**	0.0912	0.652	0.861	0.706

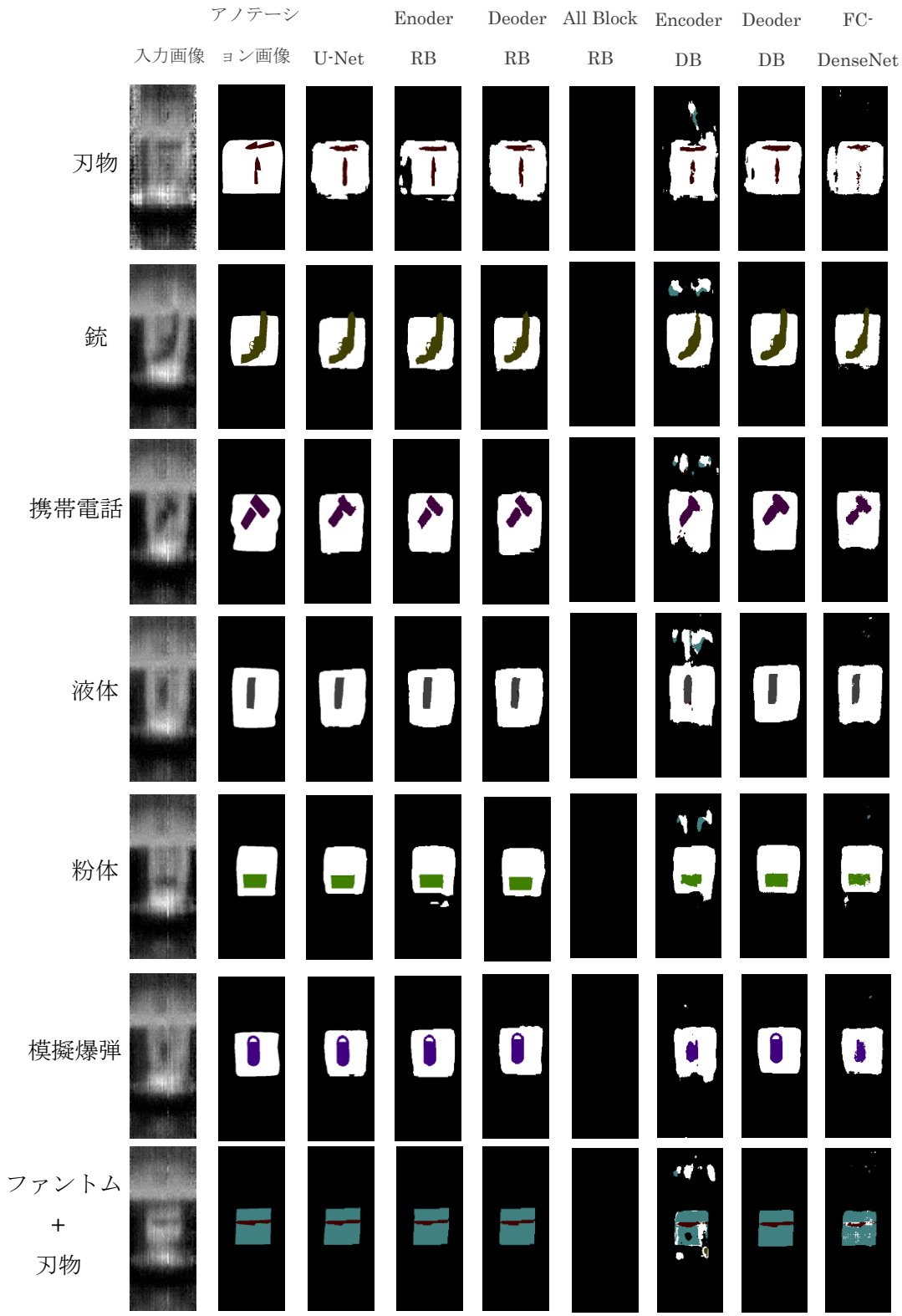

図 5.7　モデルごとの出力画像の比較

第6章　結論

6.1. まとめ

本研究では、パッシブイメージャ画像を用いた不審物検知においてデータオーギュメンテーションの工夫および既存のモデルの改良を行うことで精度を向上させることを目的とした。

データオーギュメンテーションの工夫の取り組みでは、Mixup を適用することで IoU を評価値とした精度が向上することが確認できた。また、Mixup を使用して既存の 4 種類のモデルで学習を行い IoU 値および出力画像の比較を行った際には、U-Net が最も高い mIoU を計測し不審物の外形を正確に表現できたことから、Mixup のデータオーギュメンテーションと U-Net を用いることで精度の高い検知が可能であると確認できた。

U-Net のモデル改良の取り組みでは、Residual Block および Desnse Block をデコーダに適用する際に元の U-Net のモデルと比較して IoU 値が向上した。とくに、Residual Block をデコーダに用いる際に最も高い IoU 値を計測したことからパッシブイメージャ画像中のノイズの影響を軽減するためには Residual Block を U-Net のデコーダに適用するモデル改良が有効であると確認できた。

6.2. 今後の課題

本研究における今後の課題として、新たなモデルの工夫および新たな損失関数の定義が今後の課題があげられる。

本研究において、パッシブイメージャー画像において Residual Block および Desnse Block をデコーダに適用するモデル改良が有効であると確認できた。しかし、セマンティックセグメンテーションには、本研究に用いた手法の他に Dilated Convolution や CRF などを用いる既存手法が存在する。よって、Dilated Convolution や CRF を用いてモデル改良を行うことで、更なる精度改善が可能かを検証することが 1 個の課題としてあげられる。

また、本研究において、損失関数に Categorical_CrossEntropy 用いることでを TP（True Positive）領域を最大化することで IoU 値の数値を向上させる取り組みを行った。しかし、不審物検知には、FN（False Negative）領域を最小化することにより不審物検知の漏れを防ぐと取り組みも重要である。よって、FN 領域を最小化するために新たな損失関数を定義することがもう 1 個の今後の課題としてあげられる。

謝辞

本研究は総務省より受託した「電波資源拡大のための研究開発・セキュリティ強化に向けた移動物体行動認識レーダー基盤技術の研究開発」（JPJ000254）に関する成果であり、ご支援いただいた総務省に感謝します。

また、本研究を進めるにあたり度々ご指導いただいた，指導教官である亀山渉教授ならびに菅沼睦先生に心よりお礼申し上げます。

また、本研究に取り組むにあたり度々ご協力をいただいた佐藤俊雄様、勝山裕様ならびに佐藤拓朗教授に心よりお礼申し上げます。

最後になりますが、日頃よりご助言して頂いた研究室の皆様方に心よりお礼申し上げます。

参考文献

[1]O. Ronneberger, P. Fischer, and T. Brox, ” U-Net: Convolutional Networks for Biomedical Image Segmentation,” MICCAI, pp. 234-241, 2015.

[2]H. Zhao, J. SHI, X. Qi, X. Wang and J. jia, ” Pyramid scene parsing network,” IEEE/CVPR, pp. 2881-2890, 2017.

[3]Anh. H. Dang and W. Kameyama, “Semantic Segmentation of Fashion Photos using Light-Weight Asymmetric U-Net,” IEEE/GCCE, pp. 175-178, 2019.

[4]Anh. H. Dang and W. Kameyama, "Robust Semantic Segmentation for Street Fashion Photos,” ICACT/TACT, Vol. 8, Issue 6, pp. 1248-1257, November.2019.

[5]K. HE, X. Xhang, S. Ren and J. Sun, “Deep residual learning for image recognition,” IEEE/CVPR, pp. 770-778, 2016.

[6]G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, “Densely connected convolutional networks,” IEEE/CVPR, pp. 4700-4708, 2017.

[7]S. Jégou, M. Drozdzal, D. Vazquez, A. Romero and Y. Bengio, “The one hundred layers tiramisu, “Fully convolutional densenets for semantic segmentation,” IEEE/CVPR, pp. 11-19, 2017.

[8]H. Zhang, M. Cisse, Y. N. Dauphin and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412* , 2017.

表一覧

図一覧

研究実績

	題目	発表	連盟者
研究発表	DNN モデルの違いによる 手話認識の精度比較に関する検討	2019 年 9 月 情報科学技術 フォーラム FIT2019, H-034	渡邊 滉大 亀山　渉
研究発表 （予定）	不審物検知における Mixup の適用 及び U-Net の改良 に関する検討	2021 年 3 月 電子情報通信学会 パターン認識・ メディア理解 研究会	亀山　渉 佐藤　俊雄 勝山　裕 佐藤　拓朗